

Лабораторная работа №6

Дисциплина: Архитектура компьютера

Клименко Алёна Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Символьные и численные данные в NASM	8
3.2	Выполнение арифметических операций в NASM	13
3.2.1	Ответы на вопросы по программе	18
3.3	Выполнение заданий для самостоятельной работы	19
4	Выводы	23
5	Список литературы	24

Список иллюстраций

3.1	Создание директории	8
3.2	Создание копии файла	8
3.3	Редактирование файла	9
3.4	Запуск исполняемого файла	9
3.5	Редактирование файла	10
3.6	Запуск исполняемого файла	10
3.7	Создание файла	10
3.8	Редактирование файла	11
3.9	Запуск исполняемого файла	11
3.10	Редактирование файла	12
3.11	Запуск исполняемого файла	12
3.12	Редактирование файла	13
3.13	Запуск исполняемого файла	13
3.14	Создание файла	14
3.15	Редактирование файла	15
3.16	Запуск исполняемого файла	16
3.17	Изменение программы	17
3.18	Запуск исполняемого файла	18
3.19	Создание файла	18
3.20	Запуск исполняемого файла	18
3.21	Создание файла	20
3.22	Написание программы	21
3.23	Запуск исполняемого файла	22

Список таблиц

1 Цель работы

Цель данной лабораторной работы - освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

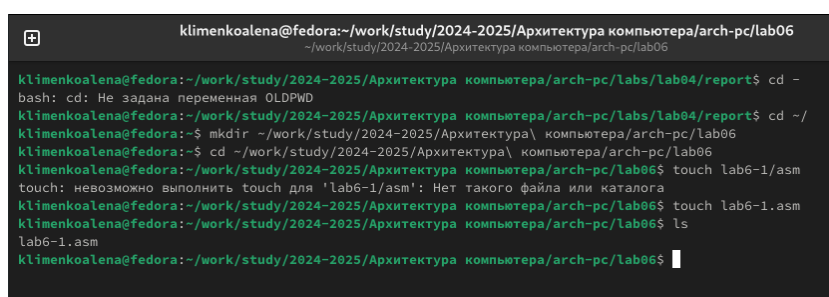
Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес операнда предоставляет место, где хранятся данные, подлежащие обработке. Это могут быть данные хранящиеся в регистре или в ячейке памяти. - Регистровая адресация – операнды хранятся в регистрах и в команде используются имена этих регистров, например: `mov ax,bx`. - Непосредственная адресация – значение операнда задается непосредственно в команде, Например: `mov ax,2`. - Адресация памяти – операнд задает адрес в памяти. В команде указывается символическое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию.

Ввод информации с клавиатуры и вывод её на экран осуществляется в символьном виде. Кодирование этой информации производится согласно кодовой таблице символов ASCII. ASCII – сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Согласно стандарту ASCII каждый символ кодируется одним байтом. Среди инструкций NASM нет такой, которая выводит числа (не в символьном виде). Поэтому, например, чтобы вывести число, надо предварительно преобразовать его цифры в ASCII-коды этих цифр и выводить на экран эти коды, а не само число. Если же выводить число на экран непосредственно, то экран воспримет его не как число, а как последовательность ASCII-символов – каждый байт числа будет воспринят как один ASCII-символ – и выведет на экран эти символы. Аналогичная ситуация происходит и при вводе данных с клавиатуры. Введенные данные будут представлять собой символы, что сделает невозможным получение корректного

результата при выполнении над ними арифметических операций. Для решения этой проблемы необходимо проводить преобразование ASCII символов в числа и обратно # Выполнение лабораторной работы

3.1 Символьные и численные данные в NASM

С помощью утилиты `mkdir` создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №6 (рис. 3.1). Перехожу в созданный каталог с помощью утилиты `cd`. С помощью утилиты `touch` создаю файл `lab6-1.asm`

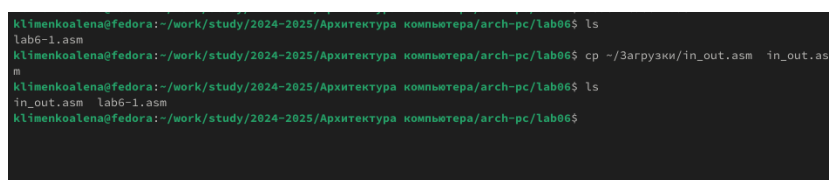


```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06

klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04/report$ cd -
bash: cd: не задана переменная OLDPWD
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04/report$ cd ~/
klimenkoalena@fedora:~$ mkdir ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
klimenkoalena@fedora:~$ cd ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-1.asm
touch: невозможно выполнить touch для 'lab6-1.asm': Нет такого файла или каталога
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-1.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ls
lab6-1.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.1: Создание директории

Копирую в текущий каталог файл `in_out.asm` с помощью утилиты `cp`, т.к. он будет использоваться в других программах (рис. 3.2).



```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ls
lab6-1.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ cp ~/Зарядки/in_out.asm in_out.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm lab6-1.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.2: Создание копии файла

Открываю созданный файл `lab6-1.asm`, вставляю в него программу вывода значения регистра `eax` (рис. 3.3).


```
GNU nano 8.1 /home/klimenkoalena/work/stud
%include 'in_out.asm'
SECTION .bss
buf1:  RESB 80
SECTION .text
Global _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF

call quit
```

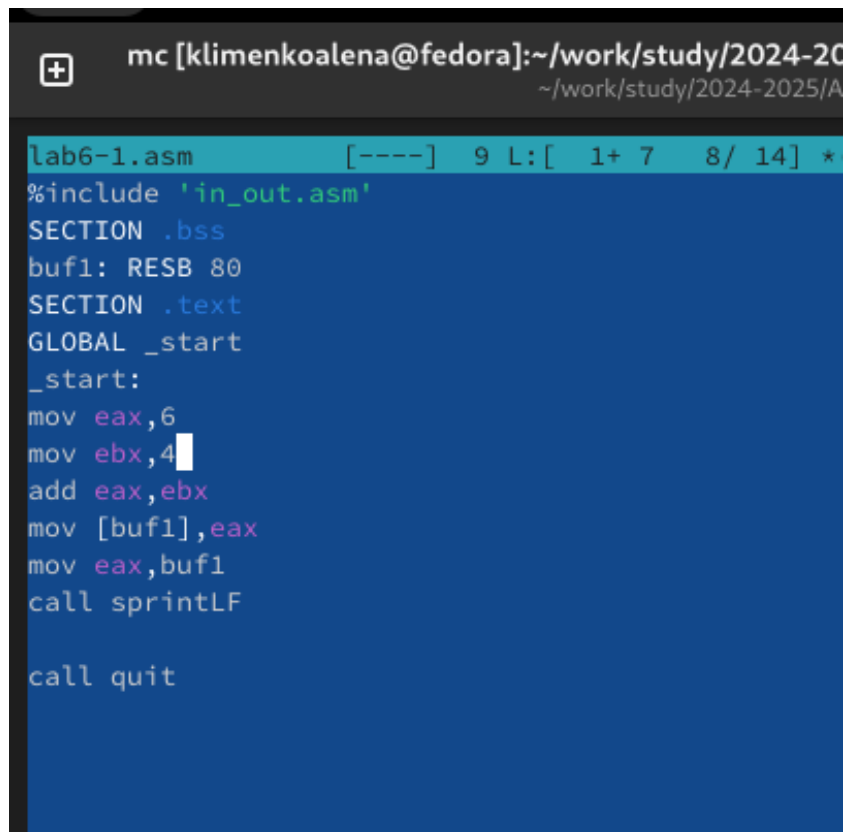
Рис. 3.3: Редактирование файла

Создаю исполняемый файл программы и запускаю его (рис. 3.4). Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6.

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-1.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-1
j
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.4: Запуск исполняемого файла

Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 3.5).

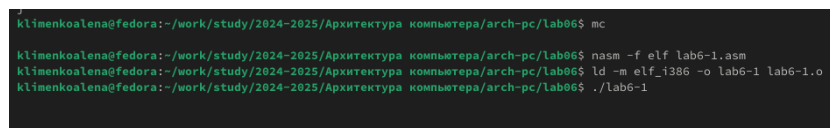


```
mc [klimenkoalena@fedora]:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
lab6-1.asm [----] 9 L:[ 1+ 7 8/ 14] *
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF

call quit
```

Рис. 3.5: Редактирование файла

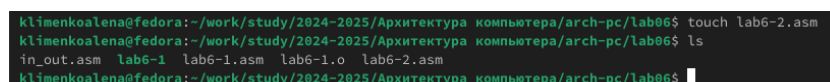
Создаю новый исполняемый файл программы и запускаю его (рис. 3.6). Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.



```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ mc
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-1.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-1
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ mc
```

Рис. 3.6: Запуск исполняемого файла

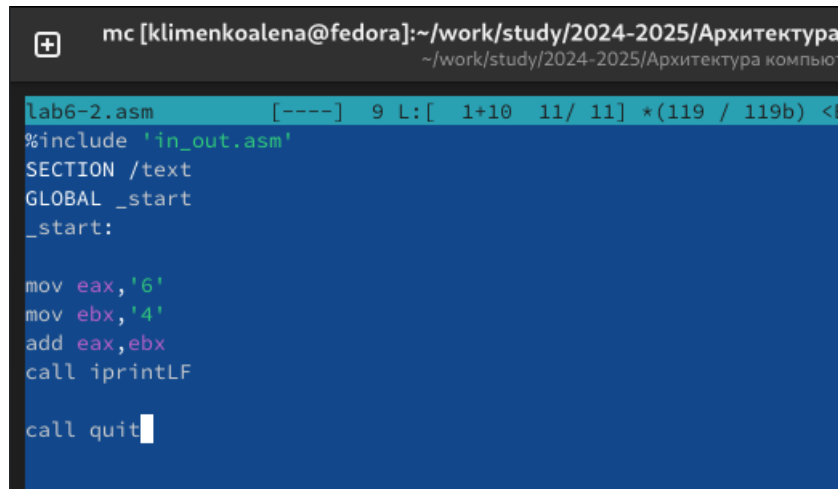
Создаю новый файл lab6-2.asm с помощью утилиты touch (рис. 3.7).



```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-2.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.7: Создание файла

Ввожу в файл текст другой программы для вывода значения регистра еах (рис. 3.8).



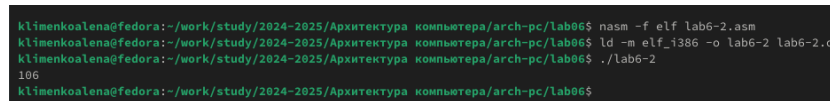
```
mc[klimenkoalena@fedora]:~/work/study/2024-2025/Архитектура
~/work/study/2024-2025/Архитектура компю
lab6-2.asm [----] 9 L: [ 1+10 11/ 11] *(119 / 119b) <E
#include 'in_out.asm'
SECTION /text
GLOBAL _start
_start:

mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF

call quit
```

Рис. 3.8: Редактирование файла

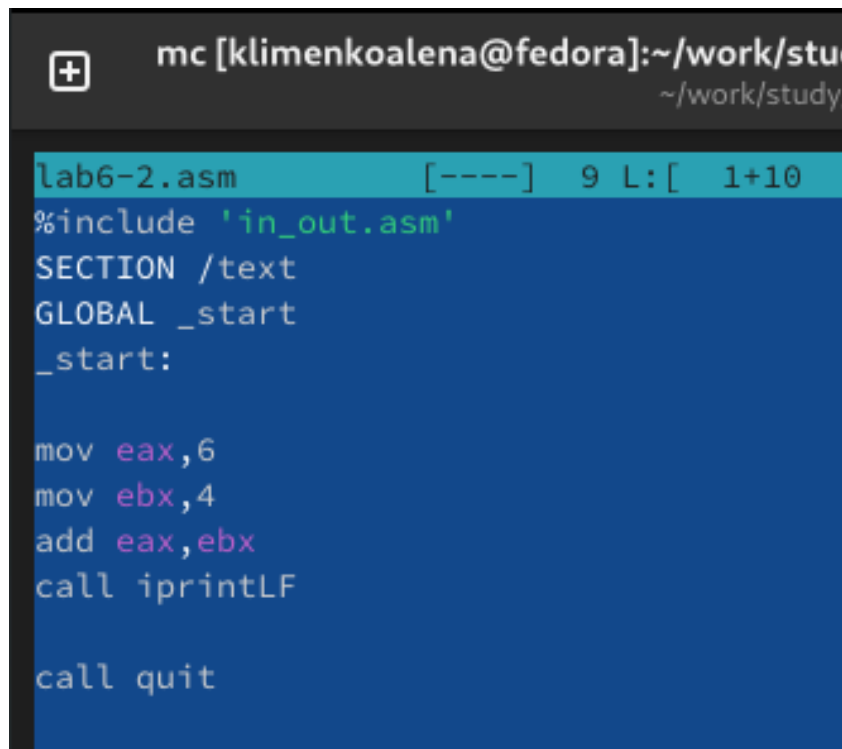
Создаю и запускаю исполняемый файл lab6-2 (рис. 3.9). Теперь вывод число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”.



```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
106
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.9: Запуск исполняемого файла

Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. 3.10).

A screenshot of a code editor window titled 'mc [klimenkoalena@fedora]:~/work/study'. The editor shows the file 'lab6-2.asm' with a line counter '9 L: [1+10'. The code content is as follows:

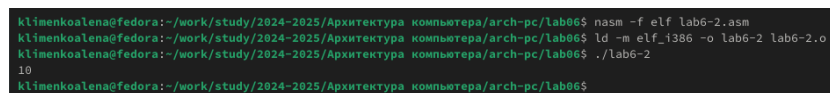
```
%include 'in_out.asm'
SECTION /text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprintLF

call quit
```

Рис. 3.10: Редактирование файла

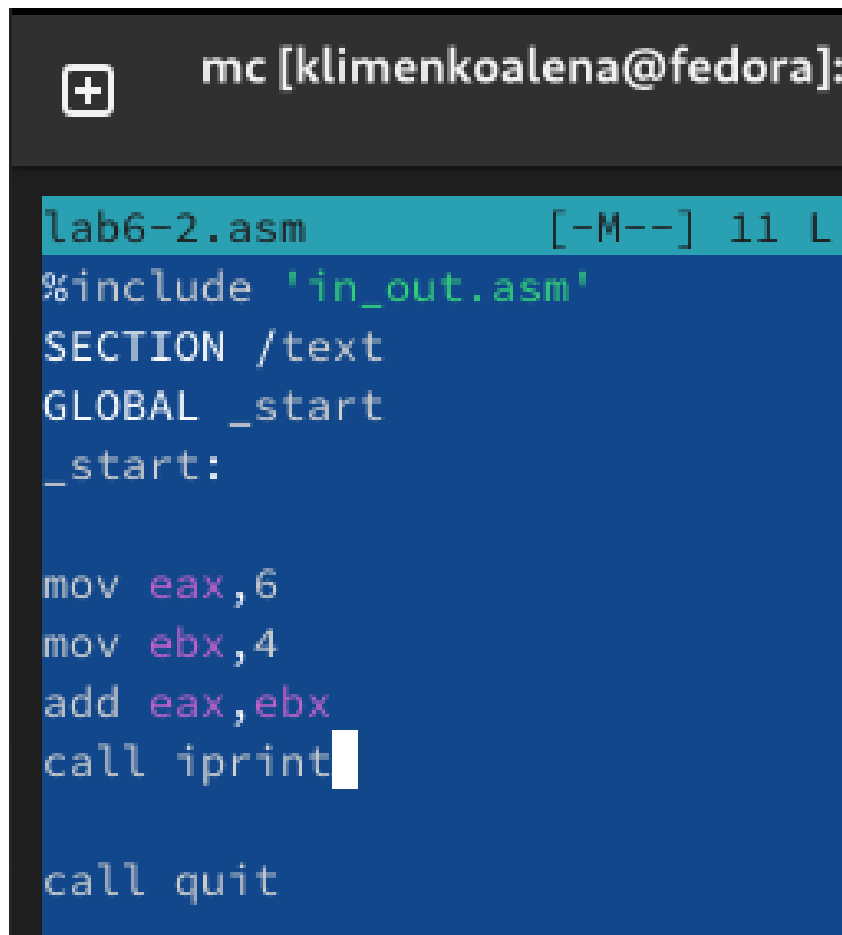
Создаю и запускаю новый исполняемый файл (рис. 3.11).. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10.

A screenshot of a terminal window showing the compilation and execution of the assembly file. The commands and their outputs are:

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
10
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.11: Запуск исполняемого файла

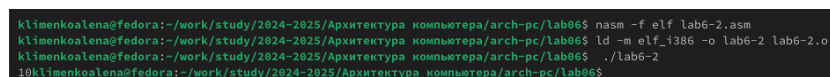
Заменяю в тексте программы функцию iprintLF на iprint (рис. 3.12).



```
mc [klimenkoalena@fedora]:  
lab6-2.asm [-M--] 11 L  
%include 'in_out.asm'  
SECTION /text  
GLOBAL _start  
_start:  
  
mov eax,6  
mov ebx,4  
add eax,ebx  
call iprint  
call quit
```

Рис. 3.12: Редактирование файла

Создаю и запускаю новый исполняемый файл (рис. 3.13). Вывод не изменился, потому что символ переноса строки не отображался, когда программа исполнялась с функцией `iprintLF`, а `iprint` не добавляет к выводу символ переноса строки, в отличие от `iprintLF`.



```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm  
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o  
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-2  
10klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.13: Запуск исполняемого файла

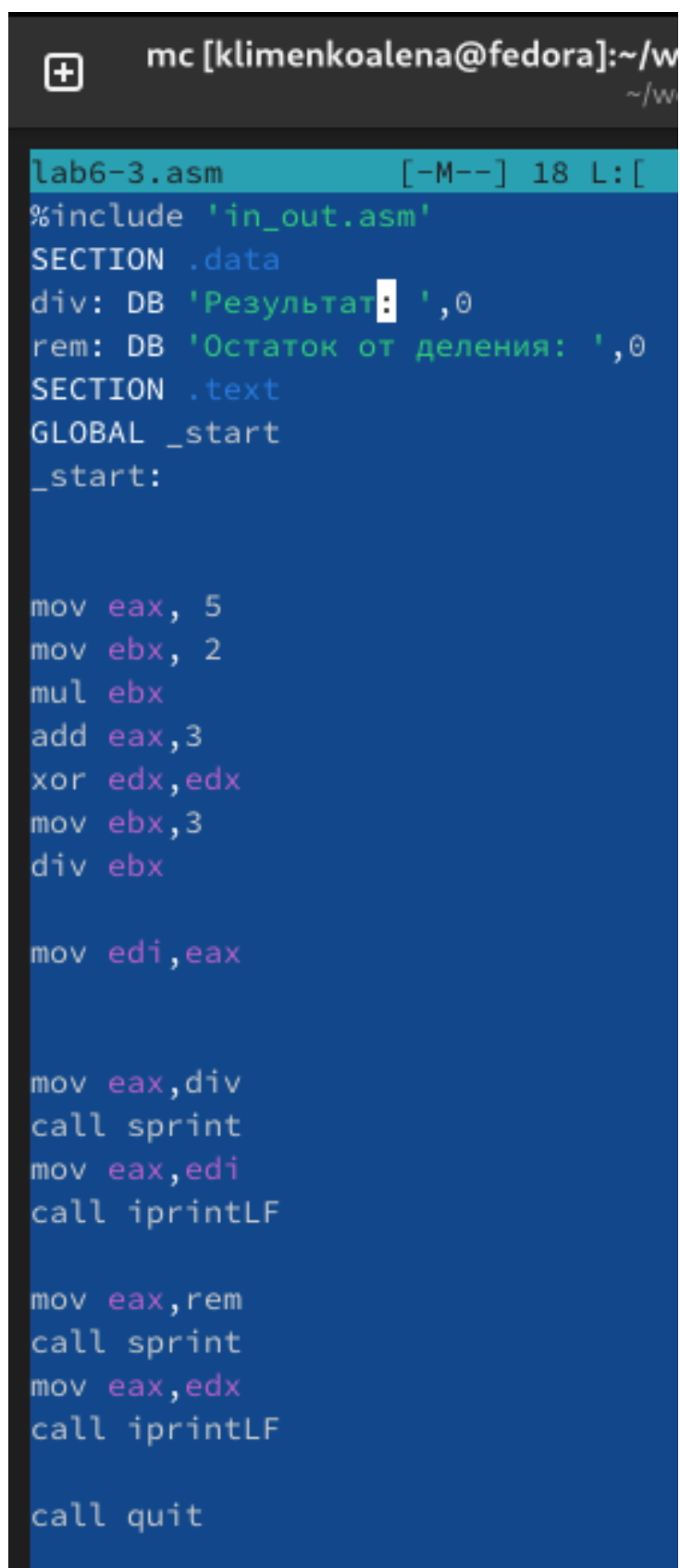
3.2 Выполнение арифметических операций в NASM

Создаю файл `lab6-3.asm` с помощью утилиты `touch` (рис. 3.14).

```
10klimenko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-3.asm
klimenko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2  lab6-2.asm  lab6-2.o  lab6-3.asm
klimenko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.14: Создание файла

Ввожу в созданный файл текст программы для вычисления значения выражения $f(x) = (5 * 2 + 3)/3$ (рис. 3.15).



```
lab6-3.asm [-M--] 18 L: [
#include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:

mov eax, 5
mov ebx, 2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit
```

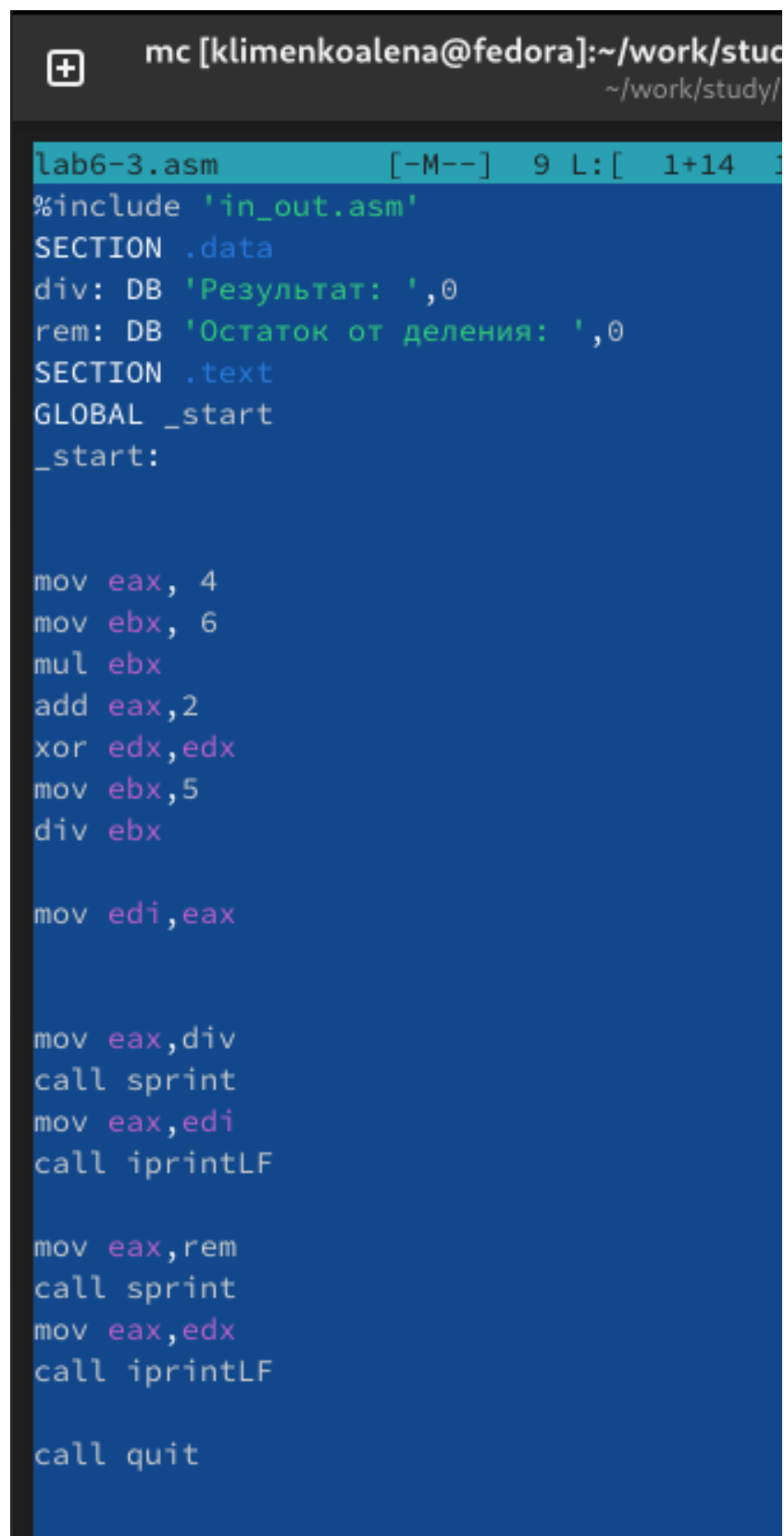
Рис. 3.15: Редактирование файла

Создаю исполняемый файл и запускаю его (рис. 3.16).

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-3.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab5-3
bash: ./lab5-3: Нет такого файла или каталога
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.16: Запуск исполняемого файла

Изменяю программу так, чтобы она вычисляла значение выражения $f(x) = (4 * 6 + 2)/5$ (рис. 3.17).



```
mc [klimenkoalena@fedora]:~/work/stud
~/work/study/

lab6-3.asm [-M--] 9 L: [ 1+14 ]
%include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:

mov eax, 4
mov ebx, 6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit
```

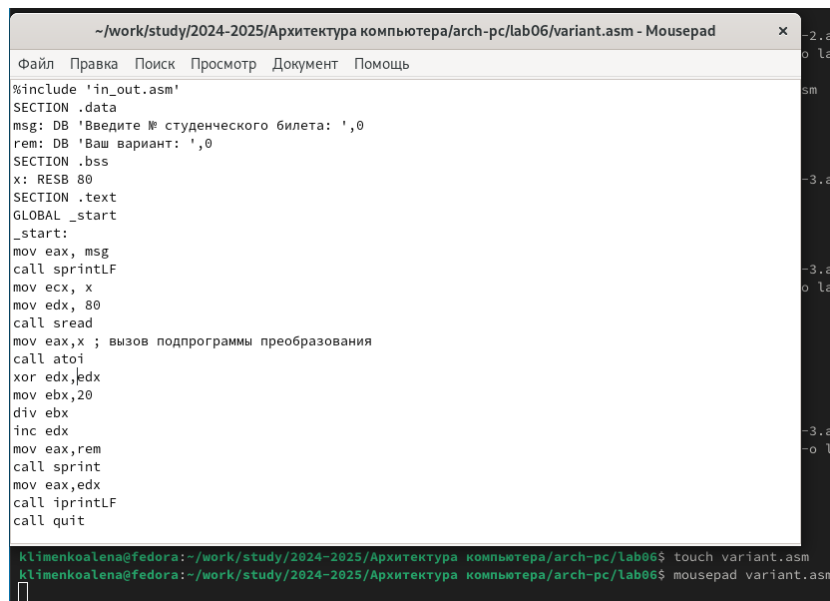
Рис. 3.17: Изменение программы

Создаю и запускаю новый исполняемый файл (рис. 3.18).

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.18: Запуск исполняемого файла

Создаю файл variant.asm с помощью утилиты touch и потом Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 3.19).



```
~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06/variant.asm - Mousepad
Файл  Правка  Поиск  Просмотр  Документ  Помощь

%include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprintf
mov eax, edx
call iprintf
call quit

klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch variant.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ mousepad variant.asm
```

Рис. 3.19: Создание файла

Создаю и запускаю исполняемый файл (рис. 3.20). Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 2.

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf variant.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246741
Ваш вариант: 2
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.20: Запуск исполняемого файла

3.2.1 Ответы на вопросы по программе

1. За вывод сообщения “Ваш вариант” отвечают строки кода:

```
mov eax,rem
```

```
call sprint
```

2. Инструкция `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`
4. За вычисления варианта отвечают строки:

```
xor edx,edx ; обнуление edx для корректной работы div
```

```
mov ebx,20 ; ebx = 20
```

```
div ebx ; eax = eax/20, edx - остаток от деления
```

```
inc edx ; edx = edx + 1
```

5. При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`
6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1
7. За вывод на экран результатов вычислений отвечают строки:

```
mov eax,edx
```

```
call iprintLF
```

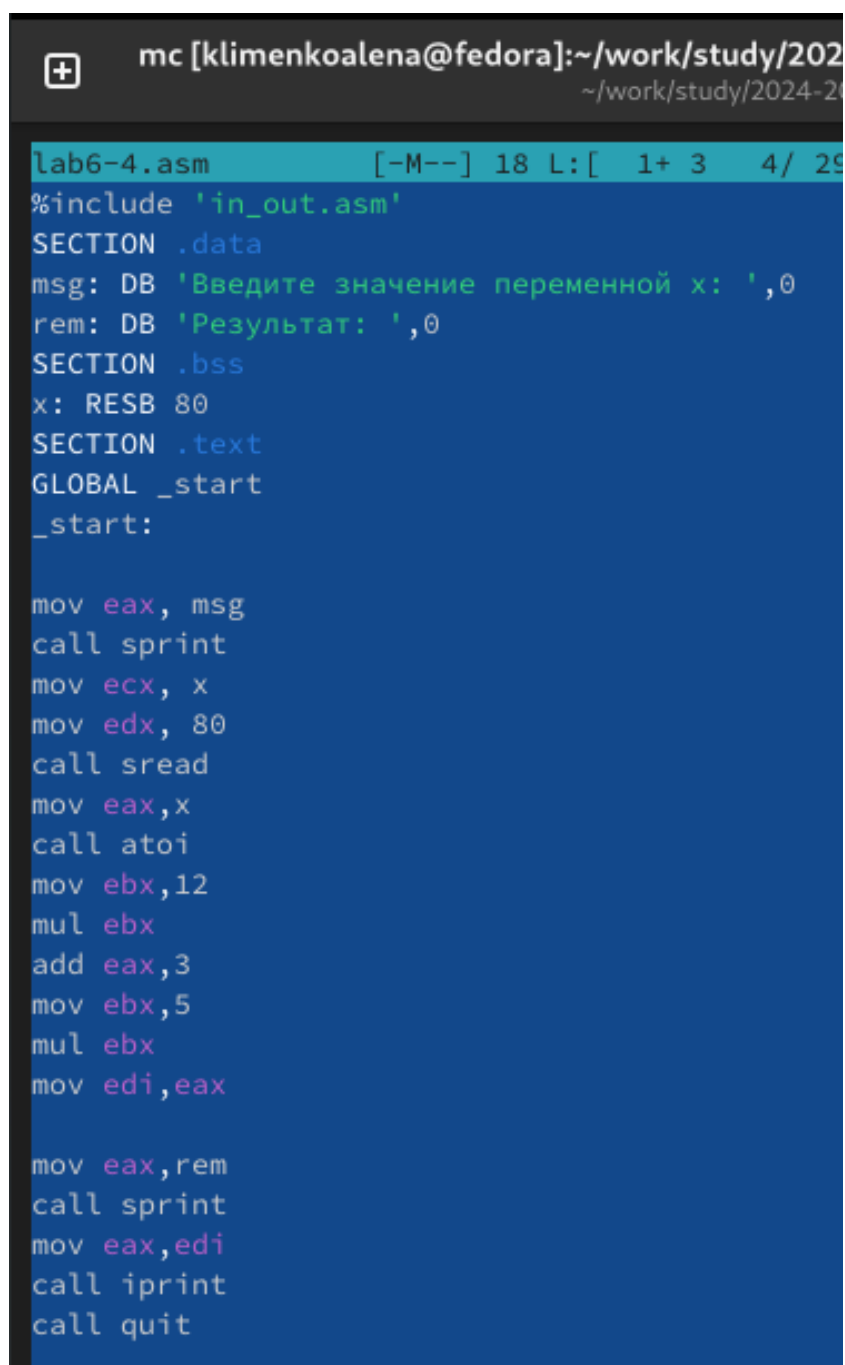
3.3 Выполнение заданий для самостоятельной работы

Создаю файл `lab6-4.asm` с помощью утилиты `touch` (рис. 3.21).

```
klimenkoalea@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-4.asm  
klimenkoalea@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.21: Создание файла

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения 2 варианта: $(12 * x + 3) * 5$ (рис. 3.22)



```
mc [klimenkoalena@fedora]:~/work/study/2024-2025
~/work/study/2024-2025
lab6-4.asm [-M--] 18 L: [ 1+ 3 4/ 29
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
mov ebx, 12
mul ebx
add eax, 3
mov ebx, 5
mul ebx
mov edi, eax

mov eax, rem
call sprint
mov eax, edi
call iprint
call quit
```

Рис. 3.22: Написание программы

Создаю и запускаю исполняемый файл (рис. 3.23). При вводе значения 1, вывод - 75, а при вводе 6, вывод - 375.

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-4.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-4
Введите значение переменной x: 1
Результат: 75klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-4
Введите значение переменной x: 6
Результат: 375klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.23: Запуск исполняемого файла

4 Выводы

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

5 Список литературы

1. ТУИС
2. Лабораторная работа №6 ::: {#refs} :::